Tema 3

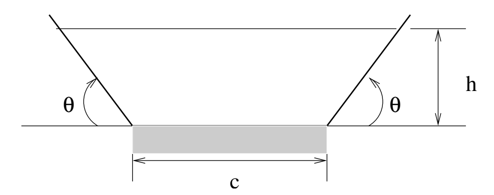
Representación gráfica 2D y 3D

Guillermo Peris Ripollés

Aplicación

Optimización del flujo de un canal de agua.

A la hora de diseñar un canal de agua, como el que se muestra en la siguiente figura, se pretende que la velocidad de flujo sea máxima.



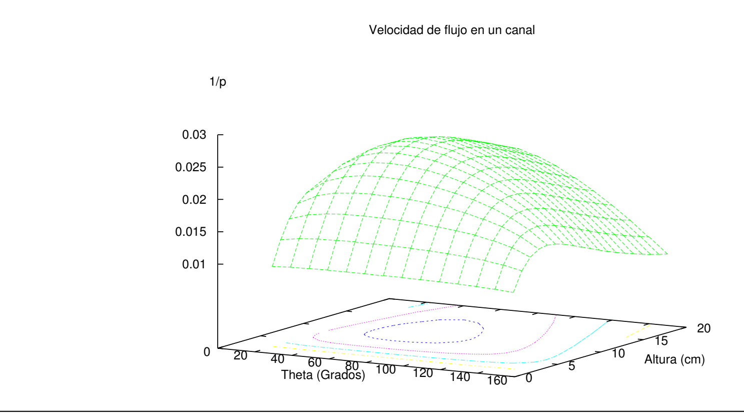
Las variables a considerar a la hora de diseñar el canal son su altura h, la anchura de la base c, y el ángulo lateral θ. Puede demostrarse que la velocidad de flujo es inversamente proporcional al perímetro, cuya expresión es:

$$p = c + \frac{2h}{\sin(\theta)}$$

mientras que el área total de flujo es

$$A = ch + h^2 cot(\theta).$$

Considerando que el canal debe tener 200 cms. de área transversal $(A = 200)$, la función $1/p$ únicamente dependerá de h y θ. Representa la superfície $1/p = f(h, \theta)$, mostrando el diagrama de contornos simultáneamente. ¿Para qué valores aproximados de h y θ la velocidad de flujo es máxima?



Contenidos

3.1. Introducción

Resulta muy habitual que los ingenieros utilicen gráficos para mostrar sus ideas de una forma más clara, ya que es más sencillo identificar tendencias en una figura que una tabla de resultados. OCTAVE dispone (junto con el paquete `octave-forge`) de un gran conjunto de funciones útiles para la creación de gráficos. En este tema estudiaremos algunas de ellas.

3.2. Representaciones 2D

Todas las funciones de que dispone OCTAVE para la creación de gráficos utilizan el programa `gnuplot`. Este es un programa que podemos usar de forma independiente de OCTAVE, aunque aquí aprenderemos a utilizarlo desde la interfaz de OCTAVE.

3.2.1. Gráficas simples

Para dibujar gráficas, OCTAVE dispone de la orden `plot(x,y)`, donde `x` e `y` son dos vectores de la misma dimensión que representan las coordenadas de las abscisas y ordenadas de los datos a representar, respectivamente. Supongamos que queremos representar la gráfica de $sin(x)$ entre 0 y 2π. Entonces, deberíamos crear en primer lugar un vector con valores de x, y el vector $sin(x)$. Para ello, podemos utilizar por ejemplo 100 puntos equiespaciados en el eje x y calcular su seno:

```
≫x = linspace(0,2*pi,100);
≫y = sin(x) ;
≫plot(x,y) ;
```

Al ejecutar este código, nos aparece una ventana similar a la siguiente:

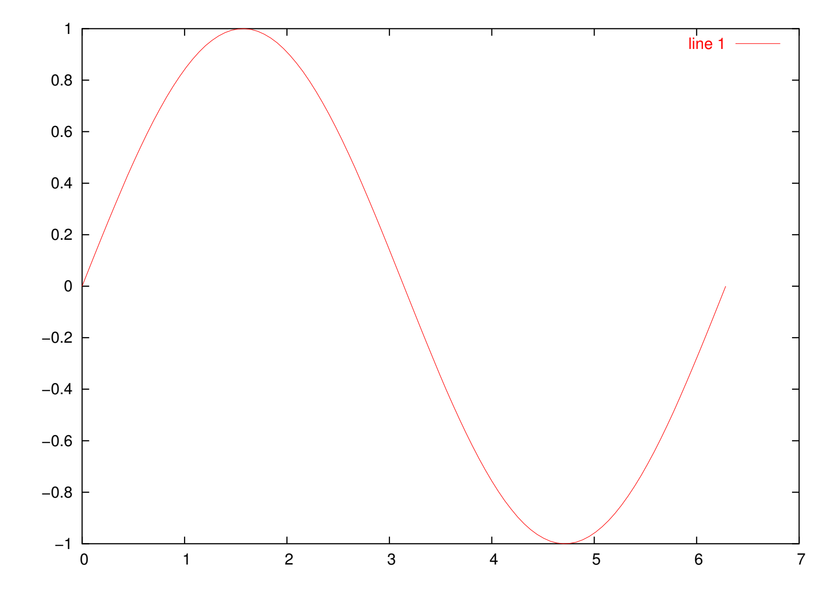


Figura 3.1: Gráfica simple.

Pensemos que cuantos más puntos utilicemos, más fiel será la representación obtenida, pero más tiempo le costará a OCTAVE obtener el resultado.

No es demasiado complicado mejorar esta gráfica. Por ejemplo, con la orden `title` se añade un título, con las órdenes `xlabel` e `ylabel` se añaden, respectivamente, etiquetas a los ejes de abscisas y ordenadas; además, la orden `grid('on')` añade una rejilla a la gráfica. Estas órdenes no surten efecto hasta que no se actualiza la gráfica con la orden `replot`. Observa el uso de estas órdenes en el siguiente ejemplo, cuyo resultado se muestra en la Figura 3.2:

```
≫title('Representacion de sin(x)'), xlabel('x'), ...
  ylabel('sin(x)'), grid('on'), replot
```

Fíjate en que los textos deben escribirse entre comillas simples.

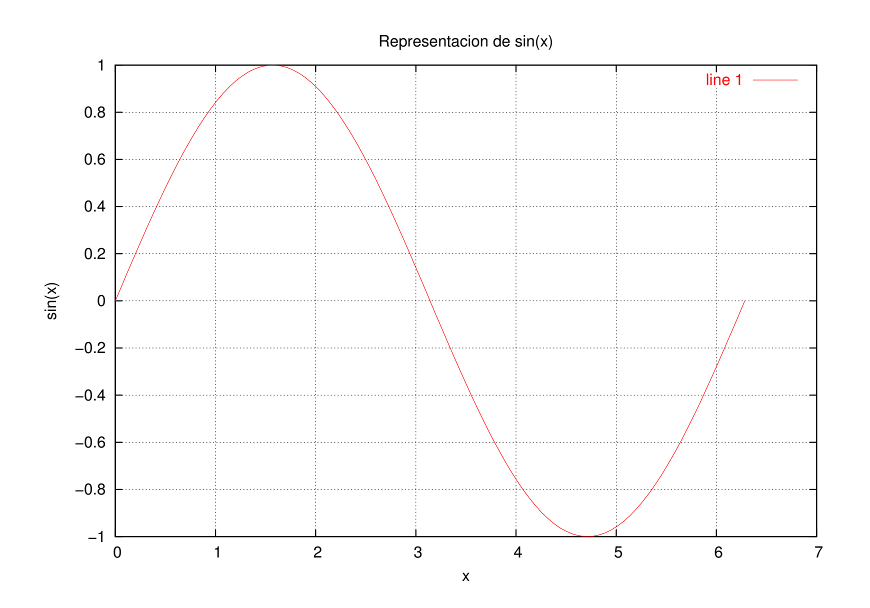


Figura 3.2: Grafica simple con título, rejilla y etiquetas.

3.2.2. Gráficas múltiples

Si estamos interesados en dibujar varias curvas en una misma gráfica, podemos hacerlo de forma simple utilizando varios pares de vectores. Por ejemplo, la orden `plot(x,y,w,z)` dibujaría en una misma gráfica las curvas de `y` en función de `x`, y `w` en función de `z`. Fíjate en el siguiente código, que dibuja las curvas $\sin(x)$ y $\cos(x)$ en la misma figura (Figura 3.3):

```
≫x = linspace(0,2*pi,100);
≫y = sin(x) ; z = cos(x);
≫plot(x,y,x,z),title('Grafica del seno y coseno de x'), ...
  xlabel('x'), grid('on') , replot
```

Octave distingue las dos líneas utilizando colores, lo cual no se puede apreciar en impresiones en blanco y negro (como la que tienes entre tus manos ;)). Sería interesante poder distinguir las curvas bien por el formato de las líneas, bien por el formato de los puntos. Para ello, debemos indicar en la orden `plot` tripletes de datos, a saber: el vector x, el vector y y un formato de lineas y puntos, teniendo en cuenta las opciones que se muestran en la Tabla 3.1. Si antecedemos un guión al indicador del

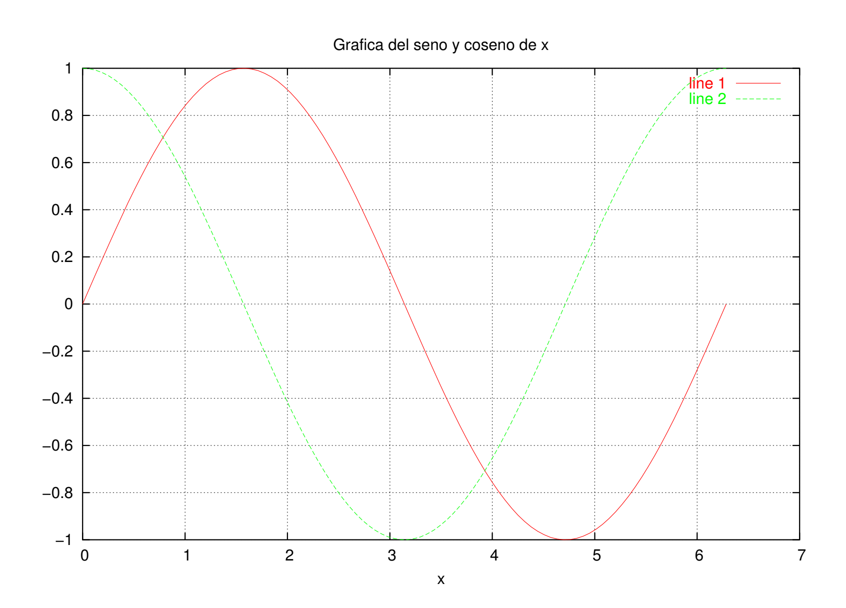


Figura 3.3: Gráfica múltiple.

punto, éstos se unen con una línea contínua[1]. Por ejemplo, una mejor representación de la curva anterior se haría con la siguiente orden (Figura 3.4):

```
≫plot(x,y,'-o',x,z,'+'),...
  title('Gráfica del seno y coseno de x'),xlabel('x'), grid('on')
, replot
```

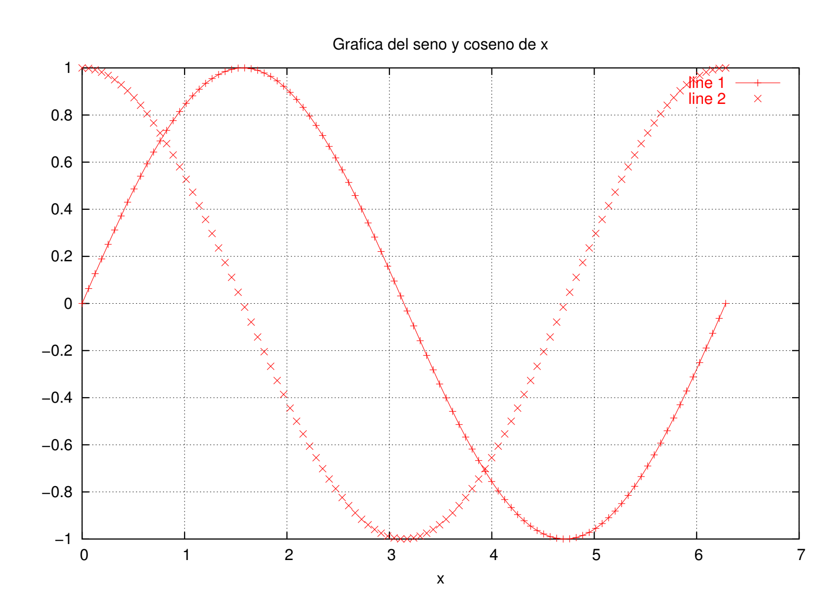


Figura 3.4: Gráfica con marcas.

Hemos comprobado que, al crear una nueva figura, esta aparece en una ventana de figuras, desapareciendo la figura anterior (si la había). Si estamos interesados en que aparezcan distintas figuras (en distintas ventanas) debemos ejecutar previamente la orden `figure`. Por último, la orden `clg` *limpia* la ventana de gráficos. Esta orden resulta conveniente utilizarla antes de crear una nueva figura.

[1] Consulta el manual de OCTAVE para obtener más información sobre el diseño de gráficas.

Punto	Indicador
punto	.
más	+
estrella	*
círculo	o
marca	x

Tabla 3.1: Opciones de marcas.

Ejercicios

▶ **1** Ejecuta las líneas de código necesarias en OCTAVE para obtener las gráficas de las figuras anteriores.

▶ **2** Dibuja la función $y = \sin(x) + x - x \cdot \cos(x)$ en el intervalo $x \in]0, 30[$. Añade un título y las etiquetas que creas oportunas.

▶ **3** A la hora de realizar una gráfica, es importante el muestreo que se realice de la función. Para entender mejor ésto, compara los siguientes ejemplos y piensa en cual es la causa de que la segunda gráfica presente un mejor aspecto que la primera.

```
≫n = 5;
≫x = 0:1/n:3;
≫y = sin(5*x) ;
≫plot(x,y)
```

```
≫n = 25;
≫x = 0:1/n:3;
≫y = sin(5*x) ;
≫plot(x,y)
```

3.2.3. Mejorando el aspecto de las gráficas

En este apartado vamos a introducir algunas órdenes más para cambiar el aspecto de los gráficos. Por ejemplo, podemos estar interesados en personalizar la leyenda explicativa de las distintas línea de una gráfica. Esta leyenda normalmente aparece con los textos `line 1`, `line 2`, etc. Para cambiar este texto, debemos escribirlo entre punto y comas (;) del formato de línea y punto de la orden `plot`. Esto se entiende mejor con el anterior ejemplo de la representación de seno y coseno:

```
≫plot(x,y,'-o;seno(x);',x,z,'.;coseno(x);')
≫title('Gráfica del seno y coseno de x'),xlabel('x')
≫grid('on') , replot
```

En ocasiones resulta interesante tener distintos gráficos en una misma ventana. Esto es lo que se conoce como **subventanas**. Para ello, podemos dividir la ventana de gráficos en dos subventanas (apiladas o contiguas) o cuatro subventanas, utilizando la orden `subplot`. Esta orden acepta 3 argumentos, donde los dos primeros indican las divisiones en filas y columnas de la ventana, y el tercero a la posición en la que se va a situar la siguiente gráfica (las ventanas se numeran de izquierda a derecha, y de arriba y abajo). La utilización de subgráficas la entenderás mejor con el siguiente ejercicio.

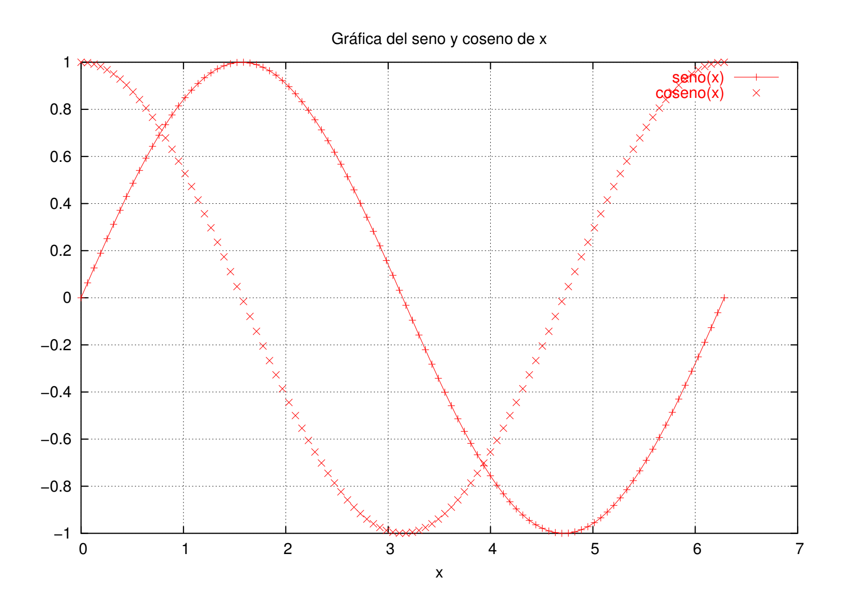


Figura 3.5: Gráfica con marcas.

Ejercicios

▶ **4** Ejecuta las siguientes lineas de código y trata de entender por ti mismo la orden `subplot`.

```
≫x = linspace(0,2*pi,100);
≫y = sin(x) ; z=cos(x);
≫subplot(2,1,1)
≫plot(x,y,';seno(x);')
≫subplot(2,1,2)
≫plot(x,z,';coseno(x);')
```

Hasta ahora hemos observado que OCTAVE fija automáticamente la escala de los ejes, ajustándola a los valores de los datos. Podemos cambiar estos valores ejecutando la orden `axis([xmin xmax ymin ymax])`, donde se indican respectivamente los valores mínimos y máximo de x, y mínimo y máximo de y.

3.3. Representaciones 3D: líneas, contornos y superfícies.

Hasta ahora hemos obtenido gráficas en dos dimensiones a partir de funciones dependientes de una sola variables. Si las funciones dependen de dos variables (o más) se necesitan representaciones de orden superior. Para ello, OCTAVE incluye distintas funciones para la representación tridimensional. A continuación vamos a explicar las funciones básicas para crear tres tipos de gráficas: líneas 3D, superficies y contornos.

3.3.1. Líneas

Se pueden representar líneas en el espacio tridimensional con la orden `plot3(x,y,z)` de OCTAVE, donde `x`, `y` y `z` son funciones de un parámetro. Por ejemplo, las siguientes

ecuaciones generan una curva en tres dimensiones a medida que varía el parámetro t:

$$\begin{aligned} x &= e^{-0.05t}sin(t) \\ y &= e^{-0.05t}cos(t) \\ z &= t \end{aligned}$$

Esta curva se puede representar con las siguientes órdenes Octave, dando lugar a la figura que se muestra a continuación:

```
≫t = [0:pi/50:10*pi];
≫plot3(exp(-0.05*t).*sin(t), exp(-0.05*t).*cos(t), t)
≫xlabel('x'), ylabel('y'), zlabel('z'), grid('on'), replot
```

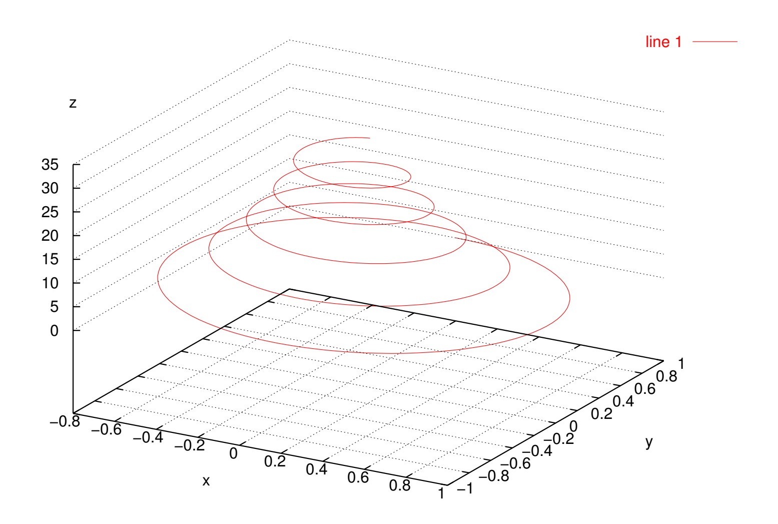


Figura 3.6: Curva 3D.

Ejercicios

► **5** Ejecuta las líneas de código anteriores en Octave para obtener la gráfica de la Figura 3.6.

3.3.2. Superficies

La función $z = f(x, y)$ representa una superficie en un sistema de coordenadas xyz. Antes de realizar la representación, es necesario crear una malla de puntos en el plano xy para calcula el valor de z en cada uno de ellos. Para ello, Octave dispone de la función `meshgrid`. La sintaxis de esta orden es:

```
[X, Y] = meshgrid(x,y)
```

donde `x` e `y` son vectores con los valores de esta variables. `X` es una matriz en la que el vector `x` se copia en cada una de sus filas, e `Y` es una matriz en la que el vector `y` se copia en cada una de sus columnas. De esta forma, podemos trabajar con las matrices `X` e `Y` para obtener una matriz `Z` en términos de la función representada. El uso de `meshgrid` se ilustra con el siguiente ejemplo sencillo:

```
≫x = [1:5] ; y = [6:10];
≫[X, Y] = meshgrid(x,y);
≫X
X =
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
≫Y
Y =
6 6 6 6 6
7 7 7 7 7
8 8 8 8 8
9 9 9 9 9
10 10 10 10 10
```

Una vez calculada la malla de valores X e Y, y con ellos la matriz Z, utilizaremos la función mesh(X,Y,Z) para crear la superficie. Veamos un ejemplo en el que se genera la superficie de la función $z = xe^{-[(x-y^2)^2+y^2]}$ para $-2 \leq x \leq 2$ y $-2 \leq y \leq 2$ con un espaciado de 0.1.

```
≫x = [-2:0.1:2]; y = x;
≫[X, Y] = meshgrid(x,y);
≫Z = X.*exp(-((X-Y.^2).^2+Y.^2));
≫mesh(X,Y,Z), xlabel('x'), ylabel('y'), zlabel('z')
```

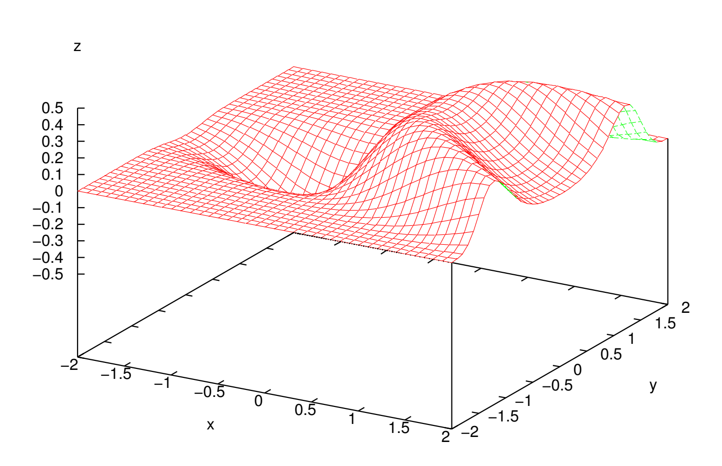


Figura 3.7: Superficie 3D.

Ejercicios

▶ **6** Ejecuta las líneas de código anteriores en OCTAVE para obtener la gráfica de la Figura 3.7.

3.3.3. Contornos

Las representaciones topográficas muestran los contornos terrestres mediante líneas de altura constante. Estas lineas se conocen como lineas de contorno. Si *caminamos* a lo largo de una de estas lineas, la altura se mantiene constante. Octave dispone de la función `contour(X,Y,X)`, que se utiliza de una foram similar a `mesh`:

```
≫x = [-2:0.1:2]; y = x;
≫[X, Y] = meshgrid(x,y);
≫Z = X.*exp(-((X-Y.^2).^2+Y.^2));
≫contour(X,Y,Z), xlabel('x'), ylabel('y'), zlabel('z')
```

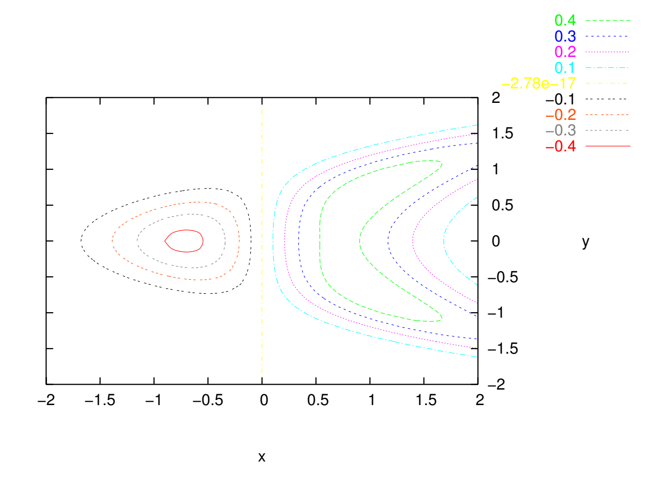


Figura 3.8: Contorno.

Las representaciones de contorno y superficie pueden combinarse con la función `meshc(X,Y,Z)`.

```
≫meshc(X,Y,Z), xlabel('x'), ylabel('y'), zlabel('z')
```

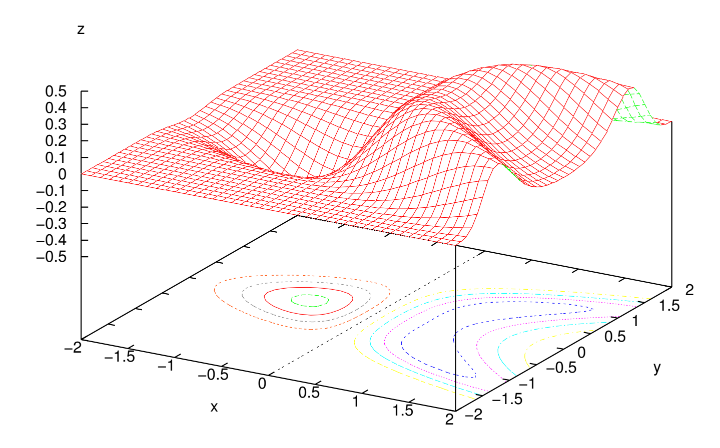


Figura 3.9: Contorno+superficie.

Ejercicios

▶ **7** Ejecuta las líneas de código necesarias en Octave para obtener las gráficas de las Figuras 3.8 y 3.9.

3.4. Aplicación

A continuación se muestra el programa que resuelve la aplicación expuesta al principio del tema. Las únicas novedades que introduce son el uso de dos nuevas funciones:

- `axis`: Esta función recibe como argumento un vector de 4 o 6 componentes, en función de que la gráfica sea bidimensional o tridimensional. Por ejemplo, en el presente caso los 6 elementos del vector serían los valores mínimos y máximos de x, seguidos de los valores mínimos y máximos de y, y por último los valores mínimos y máximos de z. La gráfica utilizará estos valores para los límites de cada uno de los ejes.
- `view`: Esta función permite cambiar el punto de vista de una función 3D. Para entender mejor su uso, ejecuta la ayuda de OCTAVE (`help view`).

Por lo demás, no presenta demasiadas complicaciones, así que trata de entenderla por ti mismo, y ejecútala en OCTAVE para comprobar que obtienes el mismo resultado.

```
% Limpiamos variables y graficos (si existen).
clear
clg

% Definicion de constantes y vectores iniciales.
A=200;
Theta = 20:10:150;
h = 2:20;

% Creamos la malla de valores (x,y) y la funcion p.
[alt,Th] = meshgrid(h, Theta);
c = (A-alt.^2./tan(Th*pi/180))./alt;
p = c + (2*alt./sin(Th*pi/180));

% Realizamos la representacion grafica.
meshc(Th,alt, 1./p)
title('Velocidad de flujo en un canal')
xlabel('Theta (Grados)'), ylabel('Altura (cm)'), zlabel('1/p')
axis([0 160 0 20 0.008 0.03])
view(30,15), replot
```

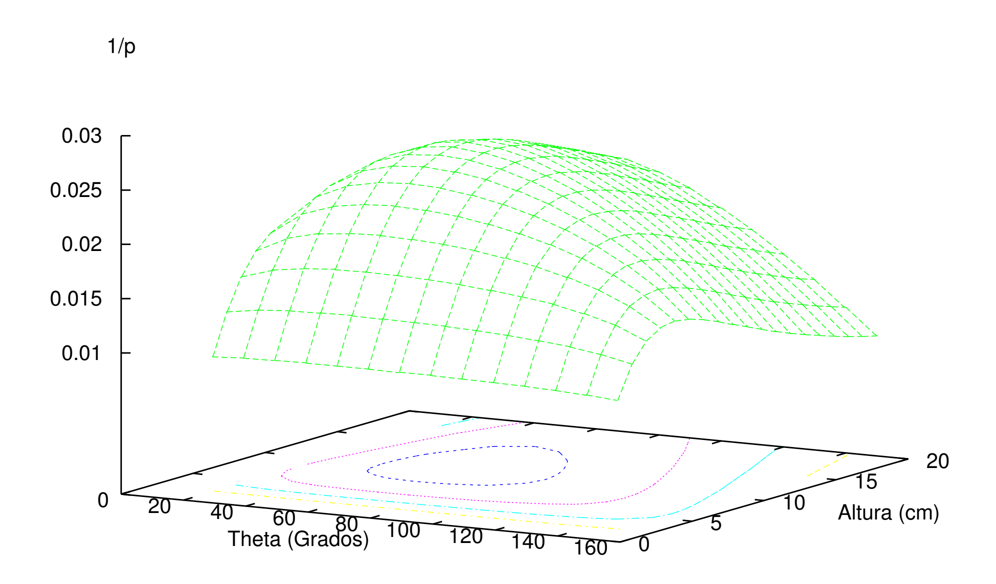


Figura 3.10: Optimización del flujo de un canal de agua.

3.5. Ejercicios prácticos

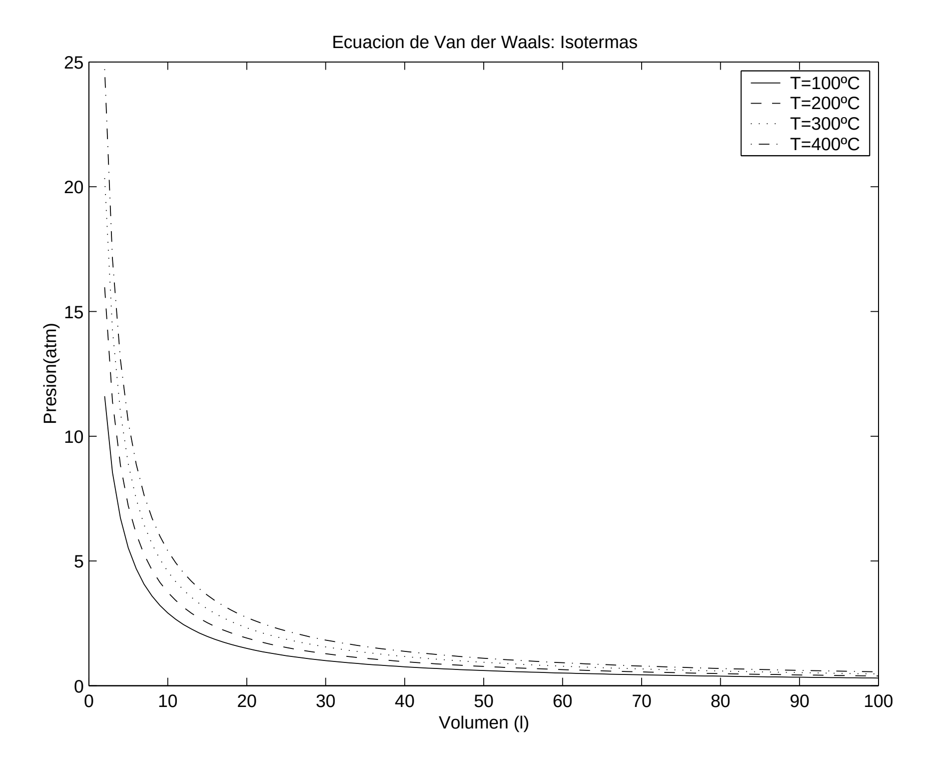
Es conveniente que pienses y realices los ejercicios que han aparecido a lo largo de la unidad marcados con el símbolo ✍ antes de acudir a la sesión de prácticas correspondiente. Deberás iniciar la sesión realizando los ejercicios marcados con el símbolo 💾. A continuación, deberás hacer el mayor número de los ejercicios siguientes.

Ejercicios

► **8** Consideremos de nuevo la ecuación de Van der Waals para un mol de gas:

$$P = \frac{RT}{V-b} - \frac{a}{V^2}$$

Escribe un programa `isotermas.m` que pida al usuario los valores de a y b, y represente una gráfica con las isotermas del gas (P en función de V, para un valor constante de T) a 100, 200, 300 y 400 grados centígrados. Recuerda que en la ecuación de Van der Waals la temperatura debe expresarse en Kelvin. Cada curva debe ir con un trazo diferenciado, con el texto que indique la isoterma que se ha representado, así como el título de la gráfica y la etiqueta de los ejes. Comprueba el funcionamiento correcto del programa con el benceno, para el cual $a = 18.78\ atm \cdot l/mol^2$ y $b = 0.1208\ l/mol$.

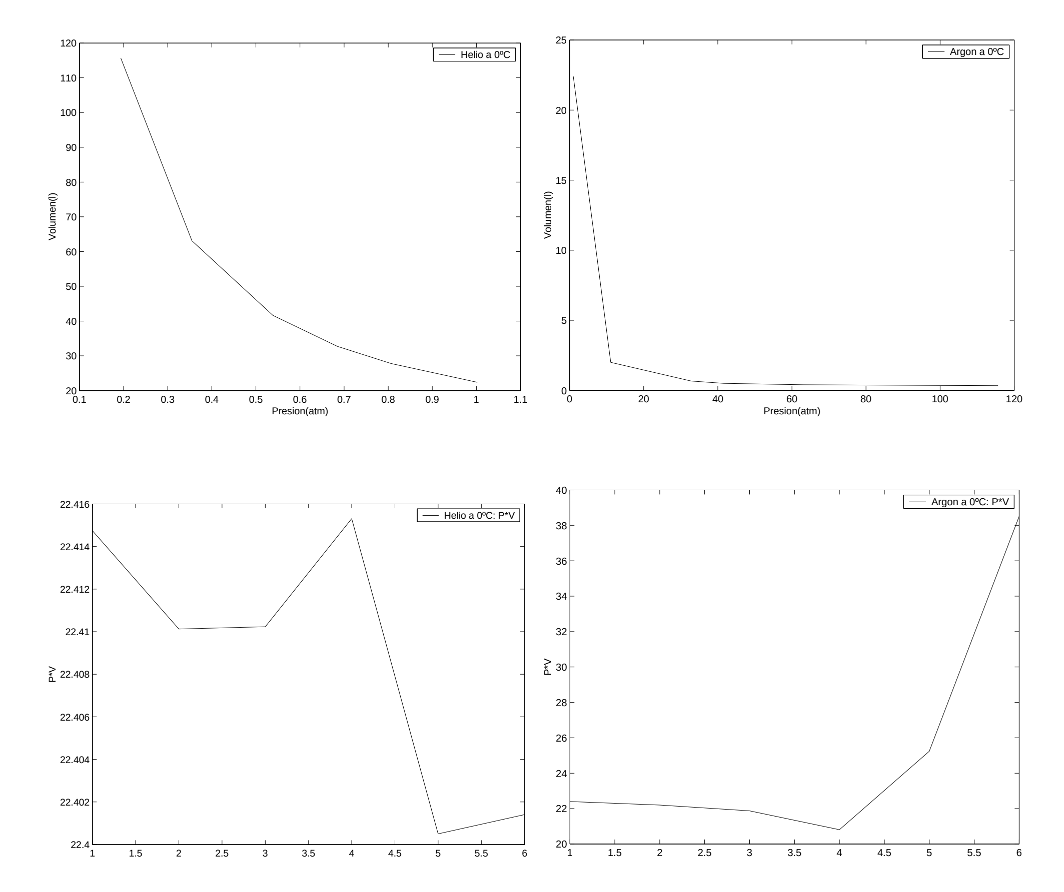


► **9** Se consideran 4.003 g. de helio y 39.944 g. de argón y se someten a cambios de presión a la temperatura de 0 grados Celsius, obteniéndose las siguientes tablas de valores:

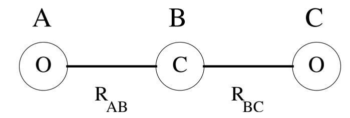
HELIO a 0º	
P(atm)	V(l)
1.002	22.37
0.8067	27.78
0.6847	32.73
0.5387	41.61
0.3550	63.10
0.1937	115.65

ARGÓN a 0º	
P(atm)	V(l)
1.000	22.4
11.10	2.000
32.79	0.667
43.34	0.500
53.68	0.400
63.68	0.333

Representa gráficamente el volumen frente a la presión en ambos gases. Comprueba que el helio es un gas que verifica la ley de Boyle-Mariotte: $P \times V = constante$, pero que el argón no cumple exactamente esta ley. Para ello, deberás representar el producto PV para cada gas. El programa se llamará `mariotte.m`.



▶ 10 Uno de los modos normales de vibración de la molécula de CO_2 consiste en la extensión lineal de los enlaces carbono-oxígeno.



Podemos obtener una aproximación a la energía de vibración de este modo normal mediante el potencial de Morse, según la siguiente ecuación:

$$E_{vib} = V_{AB}(R_{AB}) + V_{BC}(R_{BC}) \quad ,$$

donde

$$V_{XY}(R) = d_{XY}\left(e^{-2A_{XY}(R-B_{XY})} - 2e^{-A_{XY}(R-B_{XY})}\right) \quad .$$

En esta ecuación R_{XY} es la distancia entre los átomos X e Y, y el resto de parámetros para el CO_2 son: $d_{AB} = d_{BC} = 7.65eV$, $b_{AB} = b_{BC} = 1.162\AA$.

Variando los valores de R_{AB} y R_{BC}, obtén la superfície de energía potencial para este modo normal de vibración del CO_2, y la superfície de contorno asociado.

